IRIT at TREC Temporal Summarization 2015

Rafik Abbes, Bilel Moulahi, Abdelhamid Chellal, Karen Pinel-Sauvagnat, Nathalie Jane Hernandez, Mohand Boughanem, Lynda Tamine, Sadok Ben Yahia

IRIT at TREC Temporal Summarization 2015

Rafik Abbes[1], Bilel Moulahi[1,2], Abdelhamid Chellal[1], Karen Pinel-Sauvagnat[1], Nathalie Hernandez[1], Mohand Boughanem[1], Lynda Tamine[1], and Sadok Ben Yahia[2]

[1] IRIT, Paul Sabatier University
118 route de Narbonne F-31062 Toulouse cedex 9
[2] Faculty of Science of Tunisia, LIPAH, 2092 Tunis, Tunisa
{abbes, moulahi, chellal, sauvagnat, hernandez, boughanem, tamine}@irit.fr

Abstract. This paper describes the IRIT lab participation to the TREC 2015 Temporal Summarization track. The goal of the Temporal Summarization track is to develop systems that allow users to efficiently monitor information about events over time. To tackle this task, we proposed three different methods. Obtained results are presented and discussed.

1 Task description

The aim of the Temporal Summarization (TS) track is to develop systems that allow users to efficiently monitor information about events. This year, the track runs three sub-tasks that require systems to iterate over a stream corpus in a chronological order and filter relevant and novel sentences to a developing event.
We used the TREC-TS-2015F dataset provided by the track organizers [3]. Each document is identified by a *stream_id* that consists of two dash-separated parts: *timestamp* and *doc_id*. This year, 21 topics were evaluated. Each topic represents an event characterized by a *query*, a *period*, and a *type* (accident, storm, bombing, earthquake, protest, conflict). For each event, a system should emit a set of timestamped sentences called *updates* to generate the event temporal summary. The ground truth, represented by a set of *nuggets*, corresponds to a set of sentences extracted from Wikipedia by the track annotators. Matching updates to nuggets was done by track assessors. Each nugget and update are matched if they refer to the same information. To evaluate systems effectiveness, track organizers define the following metrics: the *Expected (Latency) Gain* and the *(Latency) Comprehensiveness* which are similar to the traditional IR notions of Precision and Recall (respectively).

To tackle this challenge, we propose three different approaches:

- A named entity recognition based method;
- A rank fusion based method;
- A real time summarization system relying on novelty and redundancy based approach.

This paper is organized as follows. Section 2 introduces the first method based on named entities. Section 3 describes our second method that rely on the rank aggregation approach. Section 4 presents the third method that is based on the two measures novelty and redundancy. We discuss the experimental results in Section 5. Section 6 concludes the paper.

[3] http://dcs.gla.ac.uk/ richardm/TREC-TS-2015F.tar.gz

2 Method A : Named entities recognition based method

The method presented in this section aims at retrieving from a documents stream, sentences that are relevant to a given long-running event. This method works iteratively. For each hour h, we distinguish 3 main steps : (1) selection of relevant documents using the BM25 model, (2) selection of candidate relevant sentences, and (3) verification of the novelty of candidate sentences. Novel sentences are then added to the temporal summary denoted by TS_h.

In what follows, we describe our proximity function that aims to select candidate relevant sentences as well as the novelty function.

2.1 Relevant sentence selection based on the proximity of the query terms

In this step, we analyze the sentences of the selected documents. For each sentence, we have to decide whether it is relevant or not to the target event. We rely on the intuition that a relevant sentence should be close to query Q of the event.

The proximity of a sentence with respect to Q can reflect its relevance. A sentence mentioning the event is more likely to be related to it. We express the proximity of sentence s_j with regard to the query Q using the following equation:

$$proximityScr(s_j, Q) = \frac{1}{|Q|} \sum_{t \in Q} \sum_{d=0}^{dmax} e^{-d \, * \, match(t, s_{j+d}, s_{j-d})} \tag{1}$$

$|Q|$ is the number of terms in Q, $match(t, s_x, s_y)$ is equal to 1 if t is contained in one of the sentences s_x and s_y, 0 otherwise, and $dmax$ is the maximal distance to be considered (in number of sentences). We consider only the sentences in proximity to Q by favoring those close to all of the query terms, i.e, having a $proximityScr > \tau_p$.

2.2 Novelty detection based on text divergence and named entity recognition

Sentences that are selected in the previous step could contain redundant relevant information (i.e. relevant information that has already been identified). To remove redundancy, we compare each candidate relevant sentence to all relevant sentences that are already in the temporal summary. Detecting novelty is not an easy task. Two sentences may have many terms in common, but report two different information, and inversely they may be divergent but contain the same information.

In our approach, we consider that a candidate relevant sentence s_j is novel with regard to already issued sentences (TS_h) if its text is divergent (**DIV**) and/or contains New Related Entities (**NER**), not detected in the preceding sentences TS_h. Formally s_j is novel if it fulfills the following conditions:

$$is\_novel(s_j, TS_h) = DIV(s_j, TS_h) \circ NER(s_j, TS_h) \tag{2}$$

$$DIV(s_j, TS_h) = \begin{cases} \text{false} & if \exists \;\; s_k \in TS_h, cos(s_j, s_k) > \tau_n(TS_h) \\ \text{right} & otherwise \end{cases} \tag{3}$$

$$NER(s_j, TS_h) = \begin{cases} \text{right} & if \; \exists \; x \; \in \; RE(s_j, E), \; \forall s_k \; \in \; TS_h \; x \; \notin \; RE(s_k, E) \\ \text{false} & otherwise \end{cases} \tag{4}$$

– $RE(s_j, E)$ is the set of related entities recognized in the sentence s_j [4].
– $\tau_n(TS_h)$ is a threshold for textual novelty. As the set of relevant sentences TS_h grows, the redundancy risk is higher. We thus decrease τ_n according to a Gaussian function :

$$\tau_n(TS_h) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{|TS_h|^2}{\delta^2}} \quad (5)$$

Where the σ parameter has an impact on similarity tolerance, and the δ one controls the decay rate of the threshold. $|TS_h|$ is the number of sentences in TS_h.
– The $\circ$ symbol of equation 2 can either be an ***AND*** operator to tune the system as precision-oriented by limiting redundancy or an ***OR*** operator to prioritize the exhaustivity.

2.3 Submitted runs

Table 1 presents the different configurations evaluated for this method (A).

Run	Novelty	Query
FS1A	Original Query	DIV-AND-NER
FS2A	Original Query	DIV-OR-NER
FS3A	Original Query	DIV
FS4A	Original Query + Month	DIV-AND-NER
FS5A	Original Query + Month	DIV-OR-NER
FS6A	Original Query + Month	DIV
OS1A	Original Query	DIV-AND-NER
OS2A	Original Query	DIV-OR-NER
OS3A	Original Query	DIV

Table 1: Configuration of different runs. *FS runs* are submitted in the Pre-Filtered Summarization Task. *OS runs* are submitted in the Summarization Only Task.

In column 2, we specify the query used to retrieve documents in the first step. In runs $*\{1, 2, 3\}A$, we used the the original query as given by the track organizers. We note that we selected documents containing all query terms. In runs $*\{4, 5, 6\}A$, we expanded the original query by the month(s) during which the event occurred. For example, for the topic *27: cyclone nilam* (oct, 27 2012 to nov, 2 2012), we require that the document mentions the terms *october* or *november* in addition to the query terms *cyclone* and *nilam*.
We also note, that we consider only the top 10 ranked documents in each hour retrieved using the BM25 model.

Column 3 describes the used novelty function : *DIV* means that we use only the text divergence to detect redundancy/novelty. We denote by *DIV-AND-NER* (*DIV-OR-NER*) the use of a combination of our two previously defined factors : the text divergence and the recognition of new related entities using an *AND* (*OR*) operator respectively. For the novelty threshold, we fixed δ to 100 and σ to 0.9.

For the proximity function, we fixed a strict threshold $\tau_p = 0.8$ which requires the presence of the majority of the query terms in the sentence.

[4] We use the tool developed by the *NER Stanford* group (http://nlp.stanford.edu/ner/)

3 Method B : Rank fusion based method

In this section, we present a method that is based on a temporal language modeling framework [1] and a rank aggregation scheme. This approach is designed to respond to the sub-task 2. Each query (event) term is considered as a query per se. The method includes two main steps. The first step consists in computing the single query-terms relevancy with respect to a topical matching criterion $P(w_i|d_j)$ and a temporal relevance model $P(t|w_i)$. This leads to a number of ranked lists associated with each query term. In the second step, we identify the time-span of the top K highly ranked documents of each result list and we merge the ranked lists into one ranking result. We define a set of important periods for all of these documents, that are estimated as the average of the top K document timestamps returned wrt the query terms. The goal of this step is to favour documents that are published in the same time periods as a large number of relevant documents that are returned in response to all of the query-terms.

3.1 Generating the query-terms rankings

In this step, each query-term is individually viewed as a query. The proposed model ranks documents in decreasing order of their probability of relevance based on their temporal ($P(t|w_i)$) and topical ($P(q|d)$) relevance:

$$P(d^t|w_i) = P(d,t|w_i) \propto P(d|w_i)P(t|w_i) \tag{6}$$

$$\propto P(q|w_i)P(d)P(t|w_i) \propto P(w_i|d)P(t|w_i) \tag{7}$$

Where $P(w_i|d)$ denotes the query-term likelihood on document d, $P(d)$ stands for the prior probability that d is relevant to any query-term. This temporal model is further used as a baseline. $P(w_i|d)$ is estimated using the Dirichlet smoothing, yielding:

$$P(w_i|d) = \frac{tf(w_i,d) + \mu.\frac{tf(w_i,d)}{|D|}}{|d| + \mu} \tag{8}$$

where $tf(w_i,d)$ stands for the frequency of w_i along d.

The second factor $P(t|w_i)$ conveys the relative importance of the time point t for the query-term w_i. This temporal relevance is estimated using the maximum likelihood model, which is defined as the normalized sum of the relevance scores of documents published at time t for query-term w_i:

$$P(t|w_i) = \frac{tf(w_i,D^t)}{|D^t|} \tag{9}$$

where D^t is the set of documents published at time t. This weighting function assumes the temporal independency of the query terms.

3.2 Results merging

The query-terms generated rankings obtained in the first step, give rise to different lists $r_w \in R$ wrt both topical and temporal criteria. To merge these lists, we extend an existing RRF rank fusion method [2] by injecting a temporal proximity distance that exploits the temporal term dependency. To characterize this temporal proximity, we apply the normalized variant of the so-called Gaussian kernel function. The documents scores given by the resulting model are computed as follows:

$$score(d^t \in D) = \sum_{r \in R} \frac{1}{\epsilon + r(d_t)} * kernel(t, t_{avg}) \tag{10}$$

where $r_w(d^t)$ is the position of document d in the rank list r_w and t_{avg} is the average time of the top highly ranked documents in R. We assumed that t_{avg} is the most important time period for a given query. This rewards documents, returned by all (or most of) the query terms, that are published closer to time frame of the K highly ranked documents. That is, if two documents are close enough in terms of importance and time, for all the query terms, they should be highly ranked. It is worth to mention that we assume that each relevant document identified contains at least one relevant sentence. Sentence relevance is computed using the cosine similarity measure. The Gaussian density function is computed as follows:

$$kernel(t_1, t_2) = \frac{1}{\sqrt{2\pi\sigma}} * exp[\frac{-(t_1 - t_2)^2}{2\sigma^2}] \tag{11}$$

where σ refers to the variance of the density kernel.

3.3 Submitted runs

The method presented in this section gives rise to two runs:

- *FS1B*: This run is intended to be a baseline. The computation of document relevance is based on the temporal language modeling formula given in Eq. 6. As previously mentioned, the sentence relevance are computed using the Cosine similarity measure. For each topic, we start by retrieving the first top 10 relevant documents from each hour directory using the temporal language model. We only consider documents for which scores wrt the topical criterion (Cf. Eq. 8) are higher than a threshold th_d. The latter is set to 0.5. We also define a threshold th_d, set to 0.4 to filter out non relevant sentences wrt the cosine measure. Then, for each document, we retrieve the top 5 relevant sentences. The parameter μ of the Dirichlet model is set to 2000.
- *FS2B*: This run is based on the rank fusion model presented in Eq. 10. We apply the same filtering steps used for the run *FS1B*. For each hour, we merge the first K_{fuse} results returned separately by the query terms. K_{fuse} is tuned using the TREC 2014 TS track data, and is set to 30. The rank fusion model parameters σ and ϵ are set to 300 and 10, respectively.

4 Method C : Temporal summarization based on novelty and redundancy measurement

The main purpose of this approach is to provide a short summary with maximum coverage, minimum redundancy and low latency. These requirements are fulfilled as follows: (i) The outlined approach is a fully real-time that makes select/ignore decision as soon as the sentence become available, therefore the notification time will correspond to timestamped of the document. (ii) The decision of selecting a given sentence is based on two dimensions, the novelty and redundancy. The former aims to detect new information regarding previous seen one in the stream while the later is used to avoid pushing an information already selected which keeps the summary from being redundant.

Given an event described by keywords and a stream S of sentence s_i, our approach acts like a filter where only sentences which contain at least two keywords that describe a given event are considered. An incoming sentence s_i with timestamps t_i will be added to the summary R if and only if:

$$\begin{cases} NS(s_i) >= \delta_1 = \underset{\forall\ s_j \in S^{t_i},\ t_j < t_i}{AVG} [NS(s_j)] \\ RS(s_i) >= \delta_2 = \underset{\forall\ s_j \in R^{t_i},\ t_j < t_i}{AVG} [RS(s_j)] \end{cases} \tag{12}$$

Where $NS(s_i)$ and $RS(s_i)$ are the novelty and the redundancy scores of an incoming sentence s_i. S^{t_i} and R^{t_i} are the stream and the summary at t_i (publication time of sentence s_i) respectively.

Combining this two dimensions as a conjunctive condition provides complementarity between them allowing to fulfill the requirements related to novelty, shortness and low redundancy. With linear combination, a sentence with high novelty and low redundancy scores or vice-versa, will likely be added to the summary. Also, notice here that the threshold is a parametric-free value, it is evaluated according to the previous seen values.

4.1 Novelty score

Novelty detection is generally based on similarity measures where the new document is compared to all previous seen documents or to summary only. Due to the rapid growth of the number of posted sentences in stream, similarity comparison do not fit well a real time summarization. To overcome this limit, we propose to use HybridTF-IDF [3] as a measure of novelty. The intuition behind this proposition is that a novel sentence is the one that contains a good mixture of new and important terms in the relevant sentences stream for an event. A sentence with only new terms is more likely to be a spam and irrelevant to the event of interest.

Hence, the Inverse Document Frequency (IDF) at the stream level is used as a measure of term novelty [4]. To evaluate the importance of the term within stream, we adopt the formula proposed in [3] in which the entire collection of sentences is considered as one document for computing the term frequency. Notice here that in our approach only sentences that contains keyword describing the event of interest are considered. In addition, to take into account the temporal distribution of terms in the stream, the HybridTF-IDF weight is combined with decay function. It gives a high weight to new words and those did not appear in last time window. Thereby, the novelty score of the sentence s_i with the timestamp t_i is measured as follows:

$$NS(s_i) = \sum_{w_j \in s_i} TF(w_j) \times IDF(w_j) \times Decay(w_j) \tag{13}$$

$$TF(w_j) = \frac{\#of w_j \; In All Senetnces}{\#Word In All Sentences}, \; IDF(w_j) = log_2(\frac{\#All Sentences}{\#Sentences \; w_j \; Occurs}) \tag{14}$$

$$decay(w_j) = \begin{cases} \left(\frac{\Delta t(w_j) - N}{N}\right)^2 if \Delta t(w_j) <= 2N \\ 1 \; otherwise \end{cases} \tag{15}$$

Where $\Delta t(w_j) = t^i_{w_i} - t^{i-1}_{w_i}$ represents the time since the previous occurrence of the word w_j in the stream. N represents the size of the time window.

4.2 Redundancy score

To assess the redundancy score between the new sentence regarding the summary, we propose to measure the divergence between the language model of incoming sentence and language model of each sentence in the summary. In our approach, we use the Kullback-Leibler (KL) divergence [5], in which the divergence between two sentences s_i, s_j is evaluated as follows:

$$KL(s_i, s_j) = \sum_{w_k \in s_i \cap s_j} \theta_{s_i}(w_k) \log \frac{\theta_{s_i}(w_k)}{\theta_{s_j}(w_k)} \tag{16}$$

where θ_{s_i} is the uni-gram language model of sentence s_i and $\theta_{s_i}(w_k)$ is the probability of occurrence of term w_k in sentence s_i. The incoming sentence should have a high divergence

with the most similar sentence among the summary. The latter is the one that have the minimum Kl divergence with the incoming sentence. Thereby, the redundancy score of an incoming sentence s_j is defined by the minimum KL divergence regarding each sentence in the summary R^{t_i} at time t_i as follows:

$$RS(s_i) = \min_{\forall\ s_j \in R^{t_i}} KL(s_i, s_j) \tag{17}$$

To avoid the problem of zero probabilities, we use smoothing which combines sentence model and stream model. In our runs the impact of the use of Dirichlet and Jelinek-Mercer (JM) smoothing was investigated. They are defined by the following equations respectively:,

$$\theta_s(w_j) = \frac{tf_{s_i}(w_j) \times \mu P_S^{t_i}(w_j)}{|s| + \mu} \tag{18}$$

$$\theta_{s_i}(w_j) = \lambda \times P_{s_i}(w_j) + (1 - \lambda) P_S^{t_i}(w_j) \tag{19}$$

Where λ and μ are the smoothing parameter. $P_{s_i}(w_j)$ and $P_S^{t_i}(w_j)$ are the probability of occurrence of term w_j in sentence s_i and in the stream S at time t_i respectively. They are evaluated using the maximum likelihood estimation (ML) as follows:

$$P_{T_i}(w_j) = \frac{tf_{s_i}(w_j)}{|T_i|}\ ,\ P_{S^{ti}}(w_j) = \frac{tf_{S^{t_i}}(w_j)}{|S^{ti}|} \tag{20}$$

Where $tf_{s_i}(w_j)$ and $tf_{s_i}(w_j)$ are the frequency of w_j in sentence s_i and stream S at time t_i. Smoothing parameter λ was set to 0.9 following [6] recommendation. μ was set to 100.

4.3 Submitted runs

In this approach, we assume that we have as input a relevant stream wherein filtering out irrelevant documents is not handled. Hence our participation using this approach is limited to Summarization Only sub-task. In submitted runs, the impact of the use of the decay function, the smoothing method and the threshold were investigated. Two different thresholds were tested. The first one, is parametric-free in which the threshold is defined as the average of the previous seen values in the last time window of size 300s. In the second one, thresholds were fixed to 0.27 and 3 for the novelty and redundancy score respectively. These values were learned according to experiments carried out on 2014 TREC TS filtered dataset. The parameters of each submitted run are shown in table 2.

Run	**name**	**Decay**	**Threshold δ_1 , δ_2**	**Smoothing**
OS1C	KLTFIDF-L-FIX-decay	yes	$\delta_1 = 0.27$, $\delta_2 = 3$	JM
OS2C	KLTFIDF-L-AVG-decay	yes	Average: time window of size 300 s	JM
OS3C	KLTFIDF-D-FIX-decay	yes	$\delta_1 = 0.27$, $\delta_2 = 3$	Dirichlet
OS4C	KLTFIDF-D-AVG-decay	yes	Average: time window of size 300 s	Dirichlet
OS5C	KLTFIDF-D-FIX	no	$\delta_1 = 0.27$, $\delta_2 = 3$	Dirichlet
OS6C	KLTFIDF-D-AVG	no	Average: time window of size 300 s	Dirichlet
OS7C	KLTFIDF-L-FIX	no	$\delta_1 = 0.27$, $\delta_2 = 3$	JM
OS8C	KLTFIDF-L-AVG	no	Average: time window of size 300 s	JM

Table 2: Configuration of different runs of real time summarization based on novelty and redundancy approach for Only summarization sub-task.

5 Official Results

5.1 Pre-Filtered Summarization Task

Results of our different configurations in the Pre-Filtered Summarization Task are shown in Table 3.

RunID	***nE[G]***	***nE[LG]***	***C***	***LC***	***HM(nE[LG], Lat. Comp.***
FS3A	0.0852	**0.0453**	0.5299	0.3192	**0.0754**
FS1A	0.0849	0.0414	0.4959	0.2846	0.0676
FS6A	0.0851	0.0382	0.4335	0.2137	0.0625
FS4A	**0.0875**	0.0380	0.3853	0.1909	0.0601
FS2A	0.0518	0.0251	**0.5899**	**0.3584**	0.0449
FS5A	0.0549	0.0220	0.4774	0.2368	0.0386
FS1B	0.0422	0.0140	0.2939	0.1261	0.0224
FS2B	0.0306	0.0124	0.3391	0.1563	0.0218

Table 3: Results of our configurations in the Pre-Filtered Summarization Task

Concerning the method A presented in section 2, we can see that using the BM25 model in each hour as well as the proximity function with the query terms seems to be a good criteria as we obtained respectable rates of comprehensiveness (i.e., recall) ($\in [0.38, 0.59]$). Expanding the query with the month (*FS4A, FS5A, FS6A*) degrades results especially in terms of comprehensiveness compared to runs using the *Original Query* (*FS1A, FS2A, FS3A*). The expanded query can be useful for ambiguous query like topics 36 and 37 *iraq bombing*, but seems to be too restrictive for non-ambiguous topics. For the novelty detection, combining the text divergence with the recognition of new related entities using the *"AND"* operator degrades slightly the results in terms of the measure *HM*. However, the precision is slightly improved resulting our best run in term of precision (*FS4A*). Considering the novelty using the *"OR"* operator can also be useful if the user prefers exhaustive updates (C=0.5899 for *FS2A* run).

The rank fusion method, represented by the run *FS1B*, gives low results comparing to the other runs. However, the latter slightly performs the temporal language modeling framework with a difference 2.67% in terms of the measure *HM* and 27.48% in terms of *nE[G]*. The values of precision and recall measures are low compared to the other runs, this is likely due to the topical matching model performance that fails to retrieve relevant documents. This explains the low values wrt the measure *LC* for the two runs *FS1B* and *FS2B*, that mainly rely on the language modeling framework to rerank the results (Cf., Eq. 8). We conjecture that using a good topical matching model could improve the results, as for the other runs. We also believe that a fine-grained analysis at the query level may reveals interesting insights about the types of queries that each method performs at.

5.2 Summarization Only Task

Table 4 reports the results of our participation in the sub-task 3 (Summarization Only). The best performance are obtained by the real time summarization based on the use of Hybrid-TFIDF as novelty measure and redundancy estimation using KL divergence. We observe that:

(i) the use of the decay function always improves the performance. It gives a high weight for new words in stream which improves the expected gain; (ii) the use of average as threshold outperforms the fixed threshold. In fact, using the average as threshold is more restrictive then a fixed value which leads to reduce the number of sentences pushed in summary and improves the ELG (precision). The use of fixed value as threshold brings much noise which degrades the ELG (precision). It seems that JM smoothing fits better the real time summarization of short text stream. However, it is a preliminary results and extensive experiments need to be carried out to identify which is the best smoothing method that fit well this kind of summarization task.

We re-run our filtering method presented in section 2 to tackle the Summarization Only Task. We considered only relevant documents that are detected in the top-10 results per hour. We did not use the other relevant documents although they are provided in this task. As a consequence, we missed a lot of relevant sentences. We also failed to answer some topics (32, 34 and 44). Fortunately, our method performs well in terms of precision (*nE[G]*) especially when combining the text divergence and the recognition of named entities with and *"AND'* operator.

RunID	***nE[G]***	***nE[LG]***	***C***	***LC***	***HM(nE[LG], Lat. Comp.***
OS2C	0.0595	**0.0349**	**0.6632**	0.4071	**0.0619**
OS1C	0.0524	0.0340	0.6433	0.4362	**0.0619**
OS7C	0.0523	0.0335	0.6656	0.4488	0.0614
OS8C	0.0571	0.0335	0.6642	0.4081	0.0596
OS4C	0.0536	0.0327	0.6843	0.4390	0.0591
OS6C	0.0514	0.0315	0.6779	0.4378	0.0571
OS3C	0.0434	0.0288	0.7120	0.5075	0.0538
OS5C	0.0429	0.0284	**0.7327**	**0.5202**	0.0532
OS3A	0.0820	0.0298	0.2718	0.0900	0.0422
OS1A	**0.0930**	0.0310	0.2570	0.0808	0.0420
OS2A	0.0720	0.0258	0.3029	0.0976	0.0381

Table 4: Results of our configurations in the Summarization Only Task

6 Conclusion

The experiments conducted within the Pre-Filtered Summarization Task show that the entity recognition based method gives better results than the rank aggregation based approach. We believe that a more fine-grained analysis and parameters tuning may reveal a better understanding of their performance. For the Summarization Only Task, we results show that the novelty and redundancy measurements (method C) are quite promising in generating summaries. Further studies are also needed to determine whether other dimensions could be taken into account to select relevant sentences and improve the expected gain.

References

1. Wisam Dakka, Luis Gravano, and Panagiotis G. Ipeirotis. Answering general time-sensitive queries. *IEEE Transactions on Knowledge and Data Engineering*, 24(2):220–235, 2012.
2. Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 758–759, New York, NY, USA, 2009. ACM.
3. Beaux P. Sharifi, David I. Inouye, and Jugal K. Kalita. Summarization of twitter microblogs. *The Computer Journal*, 57(3):378–402, 2014.
4. Margarita Karkali, François Rousseau, Alexandros Ntoulas, and Michalis Vazirgiannis. Using temporal IDF for efficient novelty detection in text streams. *CoRR*, abs/1401.1456, 2014.
5. S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 03 1951.
6. Arnout Verheij, Allard Kleijn, Flavius Frasincar, and Frederik Hogenboom. A comparison study for novelty control mechanisms applied to web news stories. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence, WI 2012, Macau, China, December 4-7, 2012*, pages 431–436, 2012.